D E C L A R A T I O N

I, Kazuo EDA, residing at 7 th Fl., Shuwa Kioicho Park Bldg., 3-6, Kioicho, Chiyoda-ku, Tokyo, Japan, hereby declare that I have a thorough knowledge of Japanese and English languages, and that the attached pages contains a correct translation into English of the application document of Japanese Patent Application No. 9-267373 filed on September 30, 1997 in the name of CANON KABUSHIKI KAISHA.

I further declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statement were made with the knowledge that willful false statements and the like so made, are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Signed this 24th day of January, 2007

Kazuo EDA

Translation of Japanese Patent Application No. 9-267373

[Type of Document(s)]	Application for Patent
[Reference Number]	3590011
[Filing Date]	September 30, 1997
[Addressee]	Director-General of the Patent Office, Esq.
[International Patent Classification]	H04L 12/56 H04N 1/00
[Title of The Invention]	RELAY APPARATUS, SYSTEM AND METHOD, AND STORAGE MEDIUM
[Number of Claim(s)]	18
[Inventor(s)]	
[Address/Domicile]	c/o CANON KABUSHIKI KAISHA, 30-2, Shimomaruko 3-chome, Ohta-ku, Tokyo, Japan
[Name]	Toshihiko Fukasawa
[Inventor(s)]	
[Address/Domicile]	c/o CANON KABUSHIKI KAISHA, 30-2, Shimomaruko 3-chome, Ohta-ku, Tokyo, Japan
[Name]	Hiroshi Okazaki
[Inventor(s)]	
[Address/Domicile]	c/o CANON KABUSHIKI KAISHA, 30-2, Shimomaruko 3-chome, Ohta-ku, Tokyo, Japan
[Name]	Kenichiro Tanaka
[Applicant for Patent]	
[Identification Number]	000001007
[Name]	CANON KABUSHIKI KAISHA
[Agent]	
[Identification Number]	100076428
[Patent Attorney]	
[Name]	Yasunori Ohtsuka
[Telephone Number]	03-5276-3241
[Selected Agent]	

[Identification Number]	100093908	
[Patent Attorney]		
[Name]	Kenichi Matsumoto	
[Telephone Number]	03-5276-3241	
[Selected Agent]		
[Identification Number]	100101306	
[Patent Attorney]		
[Name]	Yukio Maruyama	
[Telephone Number]	03-5276-3241	
[Detail of Fee(s)]		
[Register Number of Prepayment]	003458	
[Amount of Payment]	21000	
[List of attached documents]		
[Classification]	Specification	1
[Classification]	Drawing(s)	1
[Classification]	Abstract	1
[Number of General Power of Attorney]	9704672	
[Proof Required Y/N]	Yes	

[Type of Document] Specification

[Title of the Invention] RELAY APPARATUS, SYSTEM AND METHOD, AND STORAGE MEDIUM

[What Is Claimed Is:]

[Claim 1]

A relay apparatus for transferring information from at least one server to at least one client via a network, said server performing stream information service in its own information transmission format, said apparatus comprising:

first communication means for performing communication with said client in a communication method in correspondence with said client;

second communication means for performing communication with said server in a communication method in correspondence with said server;

first conversion means for converting a request message to said server, received via said first communication means from said client, to a request message for said server; and

second conversion means for converting information, received via said second communication means from said server, to information in a format for said client.

[Claim 2]

The relay apparatus according to claim 1, wherein said stream information service is service of moving video information obtained by predetermined image pickup means.

[Claim 3]

The relay apparatus according to claim 1, wherein said first communication means includes delivery means for executing transmission of a reply to said client in accordance with said server.

[Claim 4]

The relay apparatus according to claim 1, wherein said second communication means includes delivery means for executing transmission of a reply in accordance with said client.

[Claim 5]

The relay apparatus according to claim 1, wherein said second communication means and said first conversion means operate independently of a server to be connected.

[Claim 6]

The relay apparatus according to claim 1, wherein a protocol by said first communication means is the HTTP protocol.

[Claim 7]

A system having a plurality of relay apparatuses claimed in claim 1.

[Claim 8]

A relay apparatus, provided between a server which serves stream data via a computer network and a client, for transmitting video data from said server to said client, comprising:

message transmission/reception means for transmitting/receiving a message from another program;

message interpretation means for interpreting the message;

connection management means for managing connection with a client;

video delivery means for delivering obtained video data to a plurality of clients;

information delivery means for delivering obtained various information to a plurality of clients;

switching means for switching a method for transmitting a request message and a reply in accordance with a video server; and

execution means for executing transmission of a request message and a reply in accordance with a video server.

[Claim 9]

The relay apparatus according to claim 8, wherein said respective means operate as programs.

[Claim 10]

The relay apparatus according to claim 8, wherein said video delivery means, said information delivery means, said delivery method execution means, said message interpretation means, said connection management means, and said delivery method switching means operate as server programs different from each other.

[Claim 11]

The relay apparatus according to claim 9, further comprising message transmission/reception means for a specific video server, in addition to said message transmission/reception means.

[Claim 12]

The relay apparatus according to claim 9, wherein the number of said video delivery means, that of information delivery means, and that of delivery method execution means are more than one.

[Claim 13]

The relay apparatus according to claim 10, wherein information held by said connection management means, said video delivery means and said information delivery means are managed by using common management means such as a data-base management system.

[Claim 14]

The relay apparatus according to claim 8, wherein said video delivery means always holds latest video data.

[Claim 15]

The relay apparatus according to claim 11, wherein said information delivery means always holds latest various information.

[Claim 16]

The relay apparatus according to claim 11, wherein a remote video relay system and its client perform communication with each other by using the HTTP protocol.

[Claim 17]

A control method for controlling a relay apparatus for transferring information from at least one server to at least one client via a network, said server performing stream information service in its own information transmission format, said method comprising:

a first communication step of performing communication with said client in a communication method in correspondence with said client;

a second communication step of performing communication with said server in a communication method in correspondence with said server;

a first conversion step of converting a request message to said server, received at said first communication step from said client, to a request message for said server; and

a second conversion step of converting information, received at said second communication step from said server, to information in a format for said client.

[Claim 18]

A storage medium containing program codes, which function as a relay apparatus for transferring information from at least one server to at least one client via a network, said server performing stream information service in its own information transmission format, when read and executed by a computer, said program codes includes program codes which function as:

first communication means for performing communication with said client in a communication method in correspondence with said client;

second communication means for performing communication with said server in a communication method in correspondence with said server;

first conversion means for converting a request message to said server, received via said first communication means from said client, to a request message for said server; and

second conversion means for converting information, received via said second communication means from said server, to information in a format for said client.

[Detailed Description of the Invention]

[0001]

[Industrial Field of Utilization]

The present invention relates to a relay apparatus, system and method and storage medium, and more particularly, to a relay apparatus, system and method and storage medium for information services by a server on a network to a client.

[0002]

[Prior Art]

With high-speed and wide area Internet and Intranet environment, network application programs (application programs which run on a computer network as a platform), which conventionally handled only text data, handle multimedia data such as video and audio data having more complicated structure and requiring a larger capacity.

[0003]

In this progress, the applicant of the present invention has proposed a network application program for providing a user with a video image obtained by a

video recorder or video camera as a still image or moving image from a server program located at a remote place via a network.

[0004]

Hereinafter, the above network application program will be referred to as a "video delivery system".

[0005]

On the other hand, as World Wide Web browsers (hereinafter simply referred to as "Web browsers" or "browsers") such as Netscape Navigator (by Netscape Communications Corporation) and Internet Explorer (by Microsoft Corporation) became popular, the computer network, that has been utilized as conventional communication means only for E-mail, news service and FTP (File Transfer Protocol) and the like, is developing as the field of more various activities such as collection of information and cooperative shopping.

[0006]

Accordingly, the Web browsers have been improved so as to provide not only multimedia information display but also a general user interface for various purposes. The introduction of HTML (Hypertext Markup Language), HTTP (Hypertext Transfer Protocol), JavaScript, Java and the like and rapid improvement of their functions show this tendency.

[0007]

[Problems That the Invention Is to Solve]

The above-described development of the Web browsers has produced a need to utilize a video delivery system in the Web system by a Web browser.

[0008]

However, the video delivery system proposed by the present applicant merely uses a special-purpose client application program (referred to as a "video display client"). The special-purpose video display client is started from a Web browser. Accordingly, it is not integrated with the Web browser.

[0009]

Further, it is possible to seemingly incorporate the above video display client in the Web browser by using a technique such as Plug-in module. However, the video display client is still an independent program in the Web browser, therefore, the high freedom of design owned by the Web browser cannot be utilized. From a home page designer's standpoint, even look & feel and user interface of the video client should be changed in accordance with his/her preference or purpose. However, this need is not satisfied.

[0010]

From the above-described situation, there is a need for a video display client integrated with a Web

browser, which can be used by a Web-browser extensible language such as Java or JavaScript (by Sun Microsystems, Inc.), or generated by using such language.

[0011]

However, to satisfy this requirement, it is necessary to solve the following problems.

[0012]

1. Absorption of difference between communication methods
2. Prevention of reduction in execution efficiency
3. Absorption of difference in video delivery format

Hereinbelow, these problems will be described.

[0013]

<1. Absorption of difference between communication methods>

The basic operation of a Web browser is to transfer a file acquisition request message to a Web server, and to display data received as a reply to the request, on the premise that the data is sent in 1 : 1 correspondence with the request from the Web browser (client). Further, the Web browser must establish a communication path (hereinafter referred to as a "connection") for each request to the Web server.

[0014]

In the video delivery system, a server and a client first establish a connection therebetween, and then video information is transmitted in a one-way manner from the server. To remove this difference between these two communication methods as above, means for mediating from one communication method to the other communication method is required.

[0015]

<2. Prevention of reduction in execution efficiency>

If the video display client (Web-version video client) is integrated with a Web browser, the execution efficiency of the video display function and other functions will be reduced in comparison with the special-purpose video display client.

[0016]

In use of the special-purpose video client, it is possible to specify the client in correspondence with the target data structure and a video delivery protocol and to optimize the operation of the client for the purpose. On the other hand, in use of the Web-version client, general data processing and data display functions provided by the Web browser must be used. Nevertheless, the Web-version client will be utilized by more users than those of the special-purpose client, since the users can use the Web-version client without labor of download and installation and therefore the

Web-version client can be easily used in comparison with the special-purpose client. However, the Web-version client with low performance might lose the reputation of the video delivery system. For this reason, means for preventing the reduction in execution efficiency must be introduced.

[0017]

<3. Absorption of difference in video delivery format>

Preferably, the above requirements 1 and 2 should be satisfied as requirements to the Web-version video client, and further, the Web-version video client should be independent of specific video delivery system and video delivery method. In the Web-version video client using Java or JavaScript, as a viewer (a video display portion) is realized as a common user interface on the Web browser, video information should be displayed regardless of the difference in type of server (video server) which delivers the video image, as normal image data can be displayed regardless of its data format such as GIF and JPEG. To meet this requirement, it is necessary to provide means for absorbing the difference in video delivery format.

[0018]

[Means of solving the Problems]

The present invention has been made in consideration of the above situation, and has its

object to provide a relay apparatus, system and method and storage medium to solve the all or at least one of the above problems and enable information transmission between a server which serves information in its own communication format and a client which uses the information service on a general network, with an efficient and simple construction.

[0019]

The foregoing object is attained by providing a relay apparatus for transferring information from at least one server to at least one client via a network, the server performing stream information service in its own information transmission format, the apparatus comprising: first communication means for performing communication with the client in a communication method in correspondence with the client; second communication means for performing communication with the server in a communication method in correspondence with the server; first conversion means for converting a request message to the server, received via the first communication means from the client, to a request message for the server; and second conversion means for converting information, received via the second communication means from the server, to information in a format for the client.

[0020]

[Embodiment]

Hereinbelow, embodiments of the present invention will now be described in detail in accordance with the accompanying drawings.

[0021]

First, prior to description of the details, the outline of embodiments will be described.

[0022]

In the embodiments, the above-described three problems are solved by preparing the following function means.

[0023]

substitutional execution means for executing a function of a special-purpose client;

efficiency promotion means for efficiently enabling the substitutional execution means; and

delivery-method switching & execution means in correspondence with plural types of video delivery methods.

The above means will be described in detail below.

[0024]

<Substitutional execution means>

The substitutional execution means includes message transmission/reception means for transmitting/receiving a message from another program, message interpretation means for interpreting the

message, and connection management means for managing connection with a client.

[0025]

<Efficiency promotion means>

The efficiency promotion means includes video delivery means for delivering video data obtained from a video server to a plurality of clients, and information delivery means for delivering various information such as server status obtained from a video server to a plurality of clients.

[0026]

<Delivery-method switching & execution means>

The delivery method switching & execution means includes delivery-method switching means for switching the method for transmitting a request message and a reply in accordance with the video server, and delivery-method execution means for executing issuance of a request message and a reply in accordance with a video server.

[0027]

Hereinbelow, an example to realize the above respective means will be described as a first embodiment.

[0028]

Note that in the present embodiment, the respective elements are realized as a server client

independent of the video server and video client. The independent server program (a program which runs on a server device) will be referred to as a "conversion server".

[0029]

First, the outline of the operation of the conversion server will be described with reference to Fig. 2. Video servers 201 to 203 are programs which operate on, e.g., a general-purpose device such as a personal computer. Each general-purpose device is connected to a storage device containing video information and a video camera for obtaining a live video image. If the video information is required, the server delivers video information onto a network 207. Further, video clients 204 to 206 operate on a device such as a personal computer. Each video client is started by a user, then functions as a client, receives video data and displays the video data.

[0030]

In the video delivery system which has been proposed by the present applicant, the video server and the video client perform communication via the network 207 to perform services. For example, when the video client 204 makes a connection with the video server 201 and transmits a video acquisition request message, the video server 201 starts transmission of video data to

the video client 204. The video client 204 displays the received video data on a display or the like.

[0031]

In the present embodiment, a conversion server 208 is provided between the video servers and the video clients. The conversion server 208 converts video data transmitted from the video servers 201 to 203 via the network 207 into an HTTP message (HTTP is described in detailed in RFC (Request For Comments)) then transmits the message to all or any of the video clients 204 to 206.

[0032]

The basic operation of the present embodiment using the conversion server 208 is as follows.

[0033]

First, the video acquisition request from the video client 204 is transmitted to the conversion server 208 in place of the video server 201. The video acquisition request is transmitted in HTTP message format. The conversion server 208 converts the HTTP message into a message of a format unique to the video server 201. The converted video acquisition request message is transferred to the video server 201, and the video server 201 performs processing in accordance with the request. For example, if the content of the message is "VDOLive", the video server starts to output

a video stream, or if the content of the message is "WebView/Livescope", the video server transmits video data for only one frame.

[0034]

The result of processing by the video server is transmitted in the data format of Motion JPEG, MPEG or the like, to the conversion server 208. The conversion server 208 converts the video data into an HTTP message and sends the message to the video client 204. Note that the conversion to HTTP is made by adding information on data type, data size, date and the like to the original data.

[0035]

Next, the respective elements of the conversion server will be described with reference to Fig. 1.

[0036]

Fig. 1 shows three computers 101, 102, 103 which function as the conversion server, the video server and the video client. The computers 101, 102, 103 are interconnected via a network 104 and can mutually transmit/receive a message. Fig. 1 only shows one video client and one video server, however, actually, a plurality of video clients and video servers are used as shown in Fig. 2. Further, the number of the video servers and that of the video clients do not pose any limitation on the present invention.

[0037]

The computer 101 is provided with a secondary storage device such as a hard disk 108 for storing binary image of the conversion server. Further, the computer 101 includes a CPU 105 for actually performing the processing by the conversion server and a main memory 107 for holding an execution image (a program code and various data and table and the like used in the conversion) by the conversion server 109. The CPU 105 and the main memory 107 are connected via a bus 106.

[0038]

A conversion server 109 consists of a group of software modules as follows. Note that "software module" means a set of data, a series of procedures and a group of functions generated by a program language such as C and C++. The software module has a function or procedure as an interface for cooperation with another software module. The interfacing function or procedure is referred to as an "entry point" here. To call another module, it is necessary to somehow obtain the memory address of the entry point of the module to be called. Generally, the address information of an entry point can be obtained upon compilation or linkage time.

a message transceiver 110

a message interpreter 111

a client session manager 112

a server information manager 113

[0039]

Further, the conversion server 109 includes a processing module 114 corresponding to the type of a specific video server to communicate with the conversion server 109. Although Fig. 2 shows only one processing module 114, however, actually, a plurality of processing modules for plurality of servers may be provided.

[0040]

The processing module 114 performs processing regarding the communication method or message format unique to the specific video server to communicate with the conversion server 109. The processing module 114 further includes the following submodules:

a server session manager 115

an information deliverer 116

a video deliverer 117

a message transceiver 118 for the specific server

Next, the outline of the modules and submodules 110 to 118 will be described.

[0041]

[Message transceiver 110]

The message transceiver 110 provides other module and submodule with means for receiving a message

transmitted from the video server 102 and the video client 103 via the network 104. Further, the message transceiver 110 provides other module and submodule with means for transmitting a reply message or the like to the video server 102 and the video client 103.

[0042]

The message transceiver is realized by utilizing a general interprocess communication protocol, TCP (Transmission Control Protocol) or UDP (User Datagram Protocol) or the like using a socket or the like. Note that some type of video server requires a specific communication method or protocol such as RTP (Realtime Transfer Protocol). For such video server, a message transceiver 118 for the server is added to the processing module 114, and the function of the message transceiver 118 is utilized. The message transceiver 110 is used for communication with the video client 103 or communication with a video server which does not require a specific communication method.

[0043]

Note that the present Web browser performs communication based on the HTTP protocol on the TCP protocol, however, a different communication method might be employed in the future. In such case, a "message transceiver" for a specific video client in correspondence with the type of the client can be

prepared. The message transceiver for the client can be realized and controlled in the same way as that for the message transceiver for the server.

[0044]

[Message Interpreter 111]

The message interpreter 111 interprets various messages (hereinafter referred to as "requests" or "request messages") transmitted from a video client and instructs other modules to perform corresponding processing. Each request message includes a message ID to discriminate the request type. The messages interpreted by the message interpreter 111 are:

a session start request

a session termination request

a video acquisition request

an information acquisition request

The respective requests will be described later.

[0045]

[Client Session Manager 112]

The client session manager 112 provides means for "absorbing difference between communication methods" as described above. More specifically, for obtaining correspondence between connection with the video client 103, disconnected for each communication, and connection with the video server 102, maintained by the end of service, the client session manager 112 allots a

unique ID number to each video client upon the first connection establishment between the video client and the conversion server. Thereafter, when the video client sends a request to the conversion server 101, the ID number (connection ID) is added to the request. By this arrangement, even though the connection between the video client and the video server is disconnected for each communication, the correspondence between the video server that transmits video data and the video client that receives the video data can be maintained. The connection ID can be regarded as virtual connection which continues between the video client 103 and the conversion server 101. This virtual connection will be called a "session".

[0046]

The above processing can be easily realized by managing the correspondence between the connection ID's and corresponding video servers in the form of a table.

[0047]

[Server Information Manager 113]

The server information manager 113 is a module to manage entry points to the processing module(s) 114. For example, regarding a session start request from the video client 103, the server information manager 113 searches for an entry point of a necessary processing module for a specific server, with the type information

of the specific video server added to the request as a key. Thus, the obtained processing module takes over the control, to execute actual processing such as session starting processing.

[0048]

The above processing by the server information manager 113 can be realized by using a table containing entry points to processing modules as its values, with video-server type information (for example, a character string such as "WebView/Livescope Ver1.10") as a key.

[0049]

Further, in a case where the conversion server is realized on an OS providing dynamic link means, the processing modules may be provided as files apart from the conversion server, and necessary processing module(s) may be automatically read when starting the conversion server. The server information manager 113 also has this function. Note that this processing is not essential in the present invention. In the present embodiment, the conversion server includes necessary processing module(s) in advance.

[0050]

[Server Session Manager 115]

When the conversion server 109 receives a session start request and a session termination request from the video client 103, the server session manager 115

generates session-start and session-termination request messages unique to the video server (102 in Fig. 1), and execute session starting processing and session terminating processing to start and terminate the session with the video server.

[0051]

Note that the server session manager 115 establishes only one connection (hereinafter the connection with the video server will be referred to as a "server session") even if a plurality of video clients send a session start request to the same video server. When the server session manager 115 receives a session start request, it first checks whether or not a requested server session with a corresponding video server has been established already. If the server session has not been established yet, the server session manager 115 newly establishes a server session with the video server.

[0052]

Note that if video transfer requests to one video server are received from a plurality of clients, the conversion server perform mediation and transmit video data from the video server to the respective clients.

[0053]

[Video Deliverer 117]

When the video deliverer 117 receives a video

acquisition request from the video client 103, it issues a video acquisition request for the video server 102, and obtains latest video data. Note that if the video server periodically transmits video information, the video deliverer 117 obtains latest video data without issuing video acquisition request.

[0054]

The obtained video data is stored in a video buffer in the video deliverer 117. Regarding video acquisition requests from a plurality of video clients "approximately simultaneously received" by the conversion server 109, the video data in the video buffer can be transmitted to the video clients. In this manner, the number of communications with the video server can be reduced, and the execution efficiency can be improved.

[0055]

Note that to process the "approximately simultaneously received" video acquisition requests,

- requests arrived within a predetermined period are regarded as simultaneously arrived requests; or

- all the requests arrived from the start to the end of video acquisition request processing are regarded as simultaneously arrived requests. The process used in the embodiment will be described later.

[0056]

[Information Deliverer 116]

The video server, proposed by the present applicant, to serve video images obtained by a camera provides means for obtaining not only video data but also information on the status of the video server or information on the network 104. The video client obtains necessary information by issuing an information acquisition request to the video server.

[0057]

Further, some type of video servers have a function to automatically notify their server status even if the video client does not issue any request. For example, the server WebView/Livescope for operation of a remote video camera, automatically notifies every change in the direction and zooming of the camera to all the video clients. As the server information or notification information is common to the plurality of video clients, buffering similar to that of video data is performed.

[0058]

The information deliverer performs acquisition of server information and notification information from a video server and delivery processing by buffering. Especially, regarding a video server having a notification function, information once stored in the buffer is effective until a new change is notified from

the video server. In this case, information requests from the video clients, received during a period from a point where the server information is stored into the buffer to a point where new server information is stored into the buffer, can be processed very quickly without inquiring of the server.

[0059]

[Message Transceiver 118]

The message transceiver 118 is a transmission/reception module for a specific server. The message transceiver 118 has the same has a function similar to that of the server message transceiver 110. Further, if the processing module 114 does not require a transmission/reception module for a specific server, the message transceiver 118 is realized as a module to call the function of the server message transceiver 110.

[0060]

Next, a communication protocol between the video client 103 and the conversion server 109 will be briefly described. the video client 103 transmits the following requests to the conversion server 109 to cause the conversion server to perform necessary processing.

- Session operation

  OpenCameraServer

  CloseCameraServer

- Image acquisition

  GetLiveImage

- Information acquisition

  GetNotice

  GetVideoInfo

- Camera operation

  OperateCamera

  GetCameraControl

  GetCameraInfo

[0061]

The respective requests are based on the HTTP 1.0. protocol these requests are transmitted as GET commands generated from URL (Universal Resource Locator) format data to be described later to the conversion server. (Note that the HTTP 1.0 protocol, URL's, and the conversion from a URL to a GET command are explained in detail in the RFC 1945).

[0062]

In the present embodiment, the HTTP protocol is not necessarily used as a protocol between the video client and the video server, but a specific message method or protocol may be used. However, in such case, the video client might not be realized by a general function of the Web viewer. In use of FTP, SMTP (Simple Mail Transfer Protocol) and the like, the above problem does not occur, therefore, these protocols can

be employed for the HTTP protocol. In this case, necessary processing can also be realized by the following method. Hereinbelow, the purpose of the respective requests and URL formats of the requests, and the formats of replies to the requests will be given.

[0063]

・OpenCameraServer

[Purpose] To start session with a video server

[Format]

http://<host name of conversion server>:<port number>/OpenCameraServer?

vc_host=<host name of video server>&

vc_host=<port number of video server>&

server_type=<type of video server>

If information on the video server is necessary besides the above information (host name of video server, port number and type of video server), arbitrary information in the following format may be added.

<name of additional information>=<value of additional information>

-Example-

cc_host=<host name of camera control server>

[Reply] If processing was successful, a connection ID is returned to the video client in the following format.

HTTP/1.0 200 OK

<additional information such as data type, date and data size>

connection_id=<connection ID>

[0064]

・CloseCameraServer

[Purpose] To terminate session with a video server.

[Format]

http://<host name of conversion server>:<port number>/CloseCameraServer?

connection_id=<connection ID>

[Reply] If processing was successful, a character string "OK" is returned.

[0065]

・GetLiveImage

[Purpose] To request acquisition of video data.

[Format]

http://<host name of conversion server>:<port number>/GetLiveImage?

connection_id=<connection ID>&

frame_count=<number of frames>

At frame_count, an integer "1" or greater is set. The client can receive video data for the designated number of frames.

[0066]

[Reply] Video data for the designated number (set at

frame_count) is returned. Note that the format of reply in a case where the number of frames is "1" is somewhat different from that in a case where the number of frames is "2" since video data for two or more frames is transmitted by the Serverpush format (described in detail in the RFC 1945).

[0067]

A. In a case where the number of frames is "1"

HTTP/1.0 200 OK

<additional information such as data type, date and data size>

<video information>

B. In a case where the number of frames is "2" or greater

HTTP/1.0 200 OK

<additional information such as data type, date and data size>

Content-type:multipart/x-mixed-replace;boundary=JointServerImage--JointServerImage

Content-Length:<data length for 1 frame>

Content-type:<data type for 1 frame>

<video information for 1 frame>

--JointServerImage

...

...

--JointServerImage--

[0068]

・GetNotice

[Purpose] To obtain content of latest notification from video server.

[Format]

http://<host name of conversion server>:<port number>/GetNotice?

commenction_id=<connection ID>

[Reply] Only if the video server as the destination of server session has a function to automatically transmit a notification message, a reply in the following format can be obtained.

HTTP/1.0 200 OK

<additional information such as data type, date and data size>

...

<name of information>=<value of information>

...

For example, in case of WebView/Livescope, as the change in the direction of a video camera is notified from the video server, the following reply can be obtained:

HTTP/1.0 200 OK

...

pan=<horizontal direction of video camera>

tilt=<vertical direction of video camera>

zoom=<zooming>

The information obtained as above differs depending on the type of video server.

[0069]

・GetVideoInfo

[Purpose] To obtain video related information

[Format]

http://<host name of conversion server>:<port number>/GetVideoInfo?

connection_id=<connection ID>

[Reply] Although the type of information differs depending on the type of video server, information in the following format is returned.

HTTP/1.0 200 OK

<additional information such as data type, date and data size>

...

<name of information>=<value of information>

...

In a general video server, the following information can be obtained:

[0070]

image_width=<lateral length of video data>

image_height=<vertical length of video data>

compression_type=<compression method for video data>

frame_rate=<frame rate>

[0071]

・GetCameraInfo

[Purpose] To obtain video-camera related information.

[Format]

http://<host name of conversion server>:<port number>/GetCameraInfo?connection_id=<connection ID>

[Reply] Only if the video server can remote-operate a video camera, a reply in the following format can be obtained.

HTTP/1.0 200 OK

<additional information such as data type, date and data size>

...

<name of information>=<value of information>

...

[0072]

For example, in case of WebView/Livescope, as the change in the direction of a video camera is notified from a video server, the following reply can be obtained.

[0073]

HTTP/1.0 200 OK

...

pan_left_limit=<operation limit in leftward direction>

pan_right_limit=<operation limit in rightward direction>

pan_current_value=<current position in horizontal direction>

tilt_up_limit=<limit in upward direction>

tilt_down_limit=<limit in downward direction>

tilt_current_value=<current position in vertical direction>

zoom_wide_limit=<limit of zooming>

zoom_tele_limit=<limit of zooming>

zoom_current_value=<zoom current value>

[0074]

・GetCameraControl

[Purpose] To request camera control right.

[Format]

http://<host name of conversion server>:<port number>/GetCameraControl?connection_id=<connection ID>

[Reply] Only if the video server can remote-operate a video camera, a character string OK can be received.

[0075]

・OperateCamera

[Purpose] To request video camera operation.

[Format]

http://<host name of conversion server>:<port number>/OperateCamera?

connection_id=<connection ID>&

pan=<designation of horizontal direction>&

tilt=<designation of vertical direction>&

zoom=<designation of zooming>

[Reply] Only if the video server can remote-operate a video camera, a character string "OK" can be received.

[0076]

Next, the flow from the start to the end of session between the video client and the conversion server by using the above-described requests will be described. The flow divides into the following five phases.

[0077]

1. Start of session
2. Acquisition of video data
3. Acquisition of information
4. Camera operation
5. Termination of session

Note that the fourth phase "camera operation" is available only in a session with a camera-operating video server such as WebView/Livescope.

[0078]

1. Start of session

When the video client transmits the OpenCameraServer request to the conversion server, a session with the video server starts. As a reply to the OpenCameraServer request, a connection ID allotted to the video client is returned. The connection ID is used in all the subsequent requests.

[0079]

2. Acquisition of video data

When the video client transmits the GetLiveImage request to the conversion server, the video client can obtain video data as a reply to the request.

[0080]

3. Acquisition of information

When the video client transmits the GetVideoInfo request to the conversion server, the video client can obtain information on video size, frame rate and the like, as a reply to the request.

[0081]

4. Camera operation

Camera operation is made by the following procedure. First, the video client requests a camera operation right by the GetCameraControl request. If the video client obtains the camera operation right, the video client sends the OperateCamera request to the conversion server, and operates the camera.

[0082]

Note that the camera operation right, i.e., a right to change a camera angle (pan angle, tilt angle, zooming and the like) is owned by only one client for each camera (except a case where a plurality of cameras are connected to a video server). The camera operation right is provided in several methods already proposed by the present applicant. In this embodiment, the operation right is provided to a first-connected client for a predetermined period, prior to other clients.

[0083]

5. Termination of session

When the video client transmits the CloseCameraServer request to the conversion server, the conversion server terminates the session with the video server used by the client, and invalidates the connection ID. To obtain video data again, the video client must issue the OpenCameraServer request again to establish connection.

[0084]

Next, the operations of the respective elements in the basic operation of the conversion server will be described.

[0085]

<1. Start of session>

First, the session starting processing will be

described with reference to Fig. 3 showing the concept of operation and the flowchart of Fig. 7.

[0086]

When a video client 303 transmits the session start request (OpenCameraServer, 319) and a message transceiver 310 receives the request (step S701), the received request is transferred to a message interpreter 311 (step S702). The message interpreter 311 extracts a message ID from the transferred session start request 320 (step S703).

[0087]

The message ID in the present embodiment is a request name portion of a request except a parameter portion. In the session start request, the message ID is the character string "OpenCameraServer". The message interpreter 311 compares a group of messages pre-stored in a hard disk or the like with the received message name, and determines what request has been transmitted. As a result, if the request is not a session start request (step S704), processing with respect to another request is performed (step S705).

[0088]

If the received request is a session start request (step S704), the type, host name and port number of server, added as parameters to the session start request are extracted from the request. If

additional information unique to a specific video server is added to the request, that portion is also extracted as a pair of parameter name and value (step S706).

[0089]

Next, the message interpreter 311 issues a session start instruction (or command) (321) to a client session manager 321 (step S707). The extracted parameters are added to the session start instruction. The session start instruction (321) is realized by a message in an object-oriented language such as C++.

[0090]

The client session manager 312 receives the session start instruction (321), then first allots a connection ID to the video client 303 (step S708). The connection ID is an integer "1" or greater, and the value of the connection ID is incremented for each allotment (other methods may be employed as long as a unique ID is allotted to each client). The allotted connection ID is transmitted via the message transceiver 310 to the video client (step S709). Further, the client session manager 312 issues a processing module acquisition instruction (322) to the server information manager 313 (step S710).

[0091]

The server information manager 313 searches a

table pre-stored in a storage device for a processing module for the server (302 in Fig. 3), with the server type information added to the processing-module acquisition instruction (322) as a key (step S711). The entry point of the processing module obtained as a result of search is notified as a reply (323) to the processing-module acquisition instruction to the client session manager 312.

[0092]

Next, the client session manager 312 issues an instruction to establish a session with the server (server session start instruction, 324) via the entry point of the obtained processing module 314 to a server session manager 315 (step S712). The server session manager 315 receives the server session formation instruction (324), extracts the host name, the port number of the server and other specific additional information added to the instruction (324), searches its internal management table with these information as keys, and checks whether or not a session with the server has been established already. If the session has been established (step S713), the server session manager 315 returns the address of the session (more exactly, data structure representing the session) to the client session manager 312. The client session manager 312 adds the obtained session information to

the entry in the management table obtained with the connection ID as the key (step S714).

[0093]

Further, if the session has not been established yet (step S713), the server session manager 315 executes server -session starting processing including establishment of connection with the video server 302 (step S715) and transmission of a session start request message (327) to the video server 302 (step S716), by using the server message transceiver 316. When the session with the video server 302 is started, the information on the session (mainly including information necessary for communication with the video server such as socket) is collected and server session information is generated, and the information is returned to the client session manager 312. The client session manager 312 adds the received session information to the entry with the connection ID as the key in the internal management table.

[0094]

Then, the session starting processing ends.

[0095]

<2. Termination of session>

Next, the session termination processing will be described with reference to Figs. 4 and 8.

[0096]

When a video client 403 issues a session termination request (CloseCameraServer, 419), and a message transceiver 410 receives the request (step S801), the received request is transferred to a message interpreter 411 (step S802). The message interpreter 411 extracts a message ID from the transferred session termination request (420). The message interpreter 411 compares the message name with the message names of the respective requests, and determines what request has been transmitted (step S803). As a result, if the received request is not a session termination request (step S804), processing with respect to another request is performed (step S805).

[0097]

If the received request is a session termination request (step S804), a connection ID added as a parameter to the session termination request is extracted (step S806).

[0098]

Next, the message interpreter 411 issues a session termination instruction (421) to a client session manager 412 (step S807). The client session manager 412 receives the instruction (421), and extracts session information from its own management table with the connection ID as a key (step S808). Then, the client session manager 412 deletes the entry

from the management table (step S809).

[0099]

Then, the client session manager 412 examines how many entries with connection ID's having the value of the session information extracted at step S808 are registered in the management table (step S810). The examination may be made by counting the number of registered entries of the same session information by searching the table values, or by managing a reference count with server session information. In any case, if the session information other than the deleted entry with the connection ID (step S811) are referred to, the client session manager 412 notifies the video client (403) of the completion of the session termination processing by using the message transceiver 410 (step S815) since further processing is not necessary.

[0100]

Further, if the session information is not shared by other video client(s) (step S811), the client session manager 412 issues a server-session termination instruction (422) to a server session manager 415 (step S812). The server session manager 415 performs server-session termination processing including transmission of a session termination request (425) to a video server 402 (step S813) via a transceiver 416 for the video server 402. Finally, the server session manager

415 disconnects the connection with the video server 402, and notifies the video client 403 of the completion of the processing (step S815).

[0101]

Thus, the session termination processing is completed.

[0102]

<3. Acquisition of video data>

Next, video acquisition processing will be described with reference to Figs. 5 and 9.

[0103]

When a video client 503 issues a video acquisition request (GetLiveImage, 519), and a message transceiver 510 receives the request (step S901), the received request is transferred to a message interpreter 511 (step S902). The message interpreter 511 extracts a message ID from the transferred video acquisition request (510) (step S903). The message interpreter 511 compares the message name with message names of the respective requests to check what request has been transmitted (step S903). As a result, if the received request is not a video acquisition request (step S904), processing with respect to another processing is performed (step S905).

[0104]

If the received request is a video acquisition

request (step S904), a connection ID added as a parameter to the request is extracted (step S906). Next, the message interpreter 511 issues a video acquisition instruction (521) to a client session manager 512 (step S907). The client session manager 512 receives the instruction (521), and extracts server session information from its management table with the connection ID as a key (step S908). Further, the client session manager 512 transfers the video acquisition instruction (522) to a video deliverer 517 (step S909).

[0105]

The video deliverer 517 first examines its video data buffer managed by the video deliver 517 itself to determine whether or not the latest video data is held there. In the present embodiment, sequential numbers are allotted to the respective video data for the above examination. That is, each time video data is registered in the buffer, a number is allotted to the data. Further, the number of referred video data is managed for each video client with session information in the management table of the client session manager 512. If video data with a number greater than that of video data last referred to for the client is registered in the buffer, the video image can be regarded as the latest video image.

[0106]

If it is determined as a result of the examination of the buffer that the content of the buffer is the latest video image (step S910), the video deliverer 517 returns the video data in the buffer to the video client 403 via the message transceiver 510 (step S914). If it is determined that the content of the buffer is not the latest (step 910), the video deliverer 517 performs video acquisition processing including sending a video acquisition request 525 to a video server 502 via a message transceiver 516 for the video server 502 (step S911). When the latest video data is obtained from the video server 502 (step S912), the video deliverer 517 rewritten the content of the buffer with the new video data (step S913), and returns the video data in the buffer via the message transceiver 510 to the video client 503 (step S914).

[0107]

Thus, the video acquisition processing is completed.

[0108]

<4. Acquisition of information>

Information acquisition processing is realized by performing processing, similar to the video acquisition processing by the video deliverer in Fig. 5, by an information deliverer. Similar to the management of

video data, the latest information is managed in a buffer of the information deliverer, so that the number of communications with a video server can be reduced.

[0109]

<5. Camera operation>

In the internal processing in the conversion server, to operate a camera, a request (camera operation request) is transmitted and information on the status of the camera is returned. Therefore, the camera operation can be handled by the same method as that for the information acquisition.

[0110]

As described above, the conversion server of the present invention includes:

a message transceiver between a video client and a video server;

a message transceiver for a specific video server;

a client session manager for managing connection with a video client;

a video deliverer for obtaining video data from a video server and delivering the video data to a plurality of video client;

an information deliverer for obtaining various information from a video server and delivering the information to a plurality of video clients;

a server information manager for changing a method for issuing a request message and a reply in accordance with a video server;

a server session manager for executing issuance of a request message and a reply in correspondence with a video server.

[0111]

Accordingly, the present embodiment utilizing the HTTP protocol as a protocol between a conversion server and the video client provides video service to even a video client protected by a fire-wall system in a network.

[0112]

Further, in the present embodiment, by utilizing the HTTP GET commands in a protocol between the conversion server and the video client, a video client can obtain and display video data without extending a Web browser.

[0113]

<Second Embodiment>

Next, a second embodiment will be described as an example where the group of substitutional execution means and the group of efficiency improvement means in the above first embodiment are realized as a server independent of the group of switching means. More specifically, the processing module for a specific

server in the same server program in the first embodiment is realized as a server program independent of the conversion server. This server as the processing module will be referred to as a "subconversion server".

[0114]

First, the operation of the second embodiment will be described with reference to Fig. 10. In Fig. 10, video servers 1001 to 1003 and video clients 1004 to 1006 are interconnected via a network 1007, similar to the video servers and video clients in Fig 2. However, the second embodiment provides a plurality of conversion servers 1008 to 1010, and a plurality of subconversion servers 1011 to 1013, respectively connected to the network 1007. The video clients 1004 to 1006 perform communication with the video servers 1008 to 1010, and the video servers 1001 to 1003 establish sessions with the subconversion servers 1011 to 1013.

[0115]

In the second embodiment, even though the plurality of conversion servers are provided, the subconversion servers may be provided in correspondence with the types of video servers. For example, in a case where the subconversion server 1011 for the video server 1001 operates, the subconversion server 1011 can

be commonly used by the conversion servers 1008 to 1010 for communication with the video server 1001. Further, as the video client can use any of the conversion servers 1008 to 1010, the video client can select the most convenient conversion server in correspondence with the status of the network 1007 and the like.

[0116]

Next, the respective elements of the second embodiment will be described with reference to Fig. 6. In Fig. 6, the CPU and the main memory in Fig. 1 are omitted, however, the elements of the present embodiment operate on a computer (including a general-purpose device such as a personal computer), therefore, description will be made on the premise that a CPU, a main memory, a bus, a secondary storage device such as a hard disk are also provided in the present embodiment. Further, Fig. 6 only shows one video client 603 and one video server 602 connected via a network 620, however, actually, a plurality of video clients and video servers are connected to the network 620, as described in Fig 1.

[0117]

The respective elements of the second embodiment divide into a conversion server 604 and a subconversion server 605. The number of these servers may be increased as shown in Fig. 10.

[0118]

First, the conversion server 604 has the following modules:

a message transceiver 606;

a message interpreter 607;

a client session manager 608;

a server information manager 609;

a data base interface 610;

The modules 606 to 609 play the same roles and perform the same operations as those of the corresponding modules in Fig. 1. The data base interface 610 added to the construction of the present embodiment is a relational data base to utilize the function of a data-base management system 601. Note that the data base is of any type in the present embodiment.

[0119]

For example, the second embodiment can be realized by using an object-oriented data base as the data-base management system 601. In the client session manager and the server information manager of the first embodiment, the management of connection ID or the like is realized by using an internal table. On the other hand, data management and search functions of the second embodiment are provided by the data-base management system 601.

[0120]

Further, the server manager of the first embodiment manages entry points to processing modules for specific servers, while the server manager 609 of the second embodiment manages information such as host names, port numbers and sockets necessary for communication with the subconversion server.

[0121]

Next, the subconversion server 605 will be described. The subconversion server 605 comprises the following modules:

a message transceiver 611;

a server session manager 612;

a message transceiver 613 for a specific server

an information deliverer 614;

a video deliverer 615;

a data base interface 616;

The modules 612 to 615 play the same roles and perform the same operations as those of the corresponding modules in Fig. 1. Further, the message transceiver 611 is mainly used for communication with the conversion server 604. The message transceiver 611 has a function similar to that of the message transceiver 606 in the conversion server 604. Further, the subconversion server 605 has a data base interface 616. The contents of the buffers in the video deliverer and information deliverer are managed by the

data-base management system 601, and shared by the plurality of subconversion servers and conversion servers.

[0122]

As described above, many of the respective elements of the second embodiment have the same functions and the perform almost the same operations as those of the corresponding elements of the first embodiment.

[0123]

Accordingly, only the difference in operation from the first embodiment will be described below.

[0124]

1. Information managed by a table in the first embodiment is managed by a data base. Registration, deletion and search of data are executed by issuing a command to the data base interface.

[0125]

2. A part of instruction executed within the conversion server is transmitted/received as a request message via the network.

More specifically, such parts of instruction are as follows.

[0126]

The session generation instruction (324) and the reply to the instruction in Fig. 3.

The session termination instruction (422) and the reply to the instruction in Fig. 4.

The video acquisition instruction (522) and the replay to the instruction in Fig. 5.

[0127]

In the above-described second embodiment, the conversion server provides

a message transceiver between a video client and a video server;

a message transceiver for a specific server;

a client session manager for management of connection with a the video client;

a server information manager for switching the method of issuing a request message and a reply in accordance with a video server.

Further, the subconversion server provides

a video deliverer to obtain video data from a video server and deliver the data to a plurality of video clients;

an information deliverer to obtain various information from a video server and deliver the information to a plurality of video clients;

a server session manager for execution of issuance of a request message and a reply in accordance with a video server.

[0128]

Further, to realize the above respective modules by using the data-base management module, a data base interface is added to the conversion server and the subconversion server.

[0129]

According to the second embodiment,

- as the conversion server and the subconversion server are provided, even if a video server of a new type is added, the new video server can be utilized from all the conversion servers and video clients only by operating one subconversion server corresponding to the video server; and

- as the data management is made by utilizing the data-base management system, various information and obtained video data are shared among the conversion servers and subconversion servers. By this arrangement, even if a trouble occurs to some conversion server or subconversion server, another server can operate for the server with the trouble.

[0130]

As described above, according to the first and second embodiments, a remote video delivery system to serve a moving or still video image to a computer at a remote place via a computer network, provides

message transmission/reception means for transmitting/receiving a message from another program;

message interpretation means for interpreting the message;

connection management means for managing connection with a client;

video delivery means for delivering obtained video data to a plurality of clients;

information delivery means for delivering obtained various information to a plurality of clients;

delivery method switching means for switching the method for issuing a request message and a reply in accordance with a video server;

delivery method execution means for executing issuance of a request message and a reply in accordance with a video server.

Accordingly, the system has the following advantages.

- The difference in communication format between the video delivery system and the World Wide Web system can be absorbed.

- The reduction in execution efficiency due to integration of the video client and a Web browser can be prevented.

- The difference in video delivery method for each video server can be absorbed, and a general-purpose video client can be realized.

[0131]

Note that as described above, in the first and

second embodiments, the conversion server and the subconversion server require hardware for connection with the network, however, these servers can be realized by programs which run on general-purpose devices such as personal computers.

[0132]

Accordingly, the object of the present invention can be also achieved by providing a storage medium storing program codes for performing the aforesaid processes to a system or an apparatus, reading the program codes with a computer (e.g., CPU, MPU) of the system or apparatus from the storage medium, then executing the program.

[0133]

In this case, the program codes read from the storage medium realize the functions according to the embodiments, and the storage medium storing the program codes constitutes the invention.

[0134]

Further, the storage medium, such as a floppy disk, a hard disk, an optical disk, a magneto-optical disk, CD-ROM, CD-R, a magnetic tape, a non-volatile type memory card, and ROM can be used for providing the program codes.

[0135]

Furthermore, besides aforesaid functions according to the above embodiments are realized by executing the program codes which are read by a computer, the present invention includes a case where an OS (operating system) or the like working on the computer performs a part or entire processes in accordance with designations of the program codes and realizes functions according to the above embodiments.

[0136]

Furthermore, the present invention also includes a case where, after the program codes read from the storage medium are written in a function expansion card which is inserted into the computer or in a memory provided in a function expansion unit which is connected to the computer, CPU or the like contained in the function expansion card or unit performs a part or entire process in accordance with designations of the program codes and realizes functions of the above embodiments.

[0137]

Further, the stream information service of the present invention is applicable to any realtime information service handling operation data, audio data and the like.

[0138]

[Effect of the Invention]

As described above, according to the present invention, information transmission can be performed between a server which serves information by its own communication format and a client which receives service on a general network, with an efficient and simple construction.

[0139]

[Brief Description of the Drawings]

[Fig. 1] This figure is a block diagram showing the construction of a system according to a first embodiment of the present invention;

[Fig. 2] This figure is a schematic diagram showing the connection relation of apparatuses in the system of the first embodiment;

[Fig. 3] This figure is a block diagram explaining the operation of session starting processing according to the first embodiment;

[Fig. 4] This figure is a block diagram explaining the operation of session terminating processing according to the first embodiment;

[Fig. 5] This figure is a block diagram explaining the operation of video acquisition processing according to the first embodiment;

[Fig. 6] This figure is a block diagram showing the construction of the system according to a second embodiment of the present invention;

[Fig. 7] This figure is a flowchart showing the operation procedure of the session starting processing according to the second embodiment;

[Fig. 8] This figure is a flowchart showing the operation of the session terminating processing according to the second embodiment;

[Fig. 9] This figure is a flowchart showing the operation of the video acquisition processing according to the second embodiment;

[Fig. 10] This figure is a schematic diagram showing the connection relation of the apparatuses in the system according to the second embodiment.

[Type of the Document] Abstract

[Abstract]

[Object]

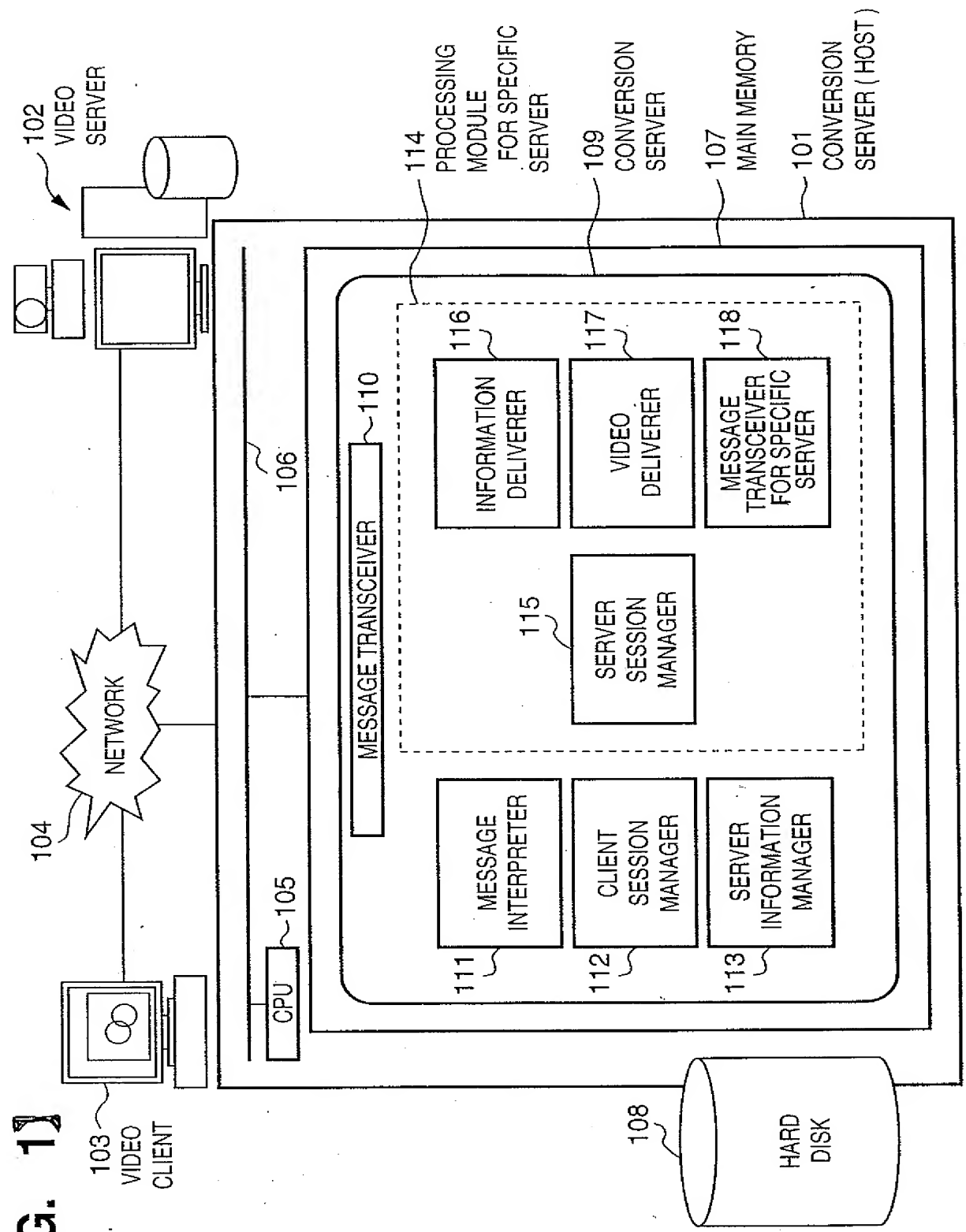
The present invention enables information transmission between a server which serves information by its own communication format and a client which receives service on a general network, with an efficient and simple structure.

[Means of solving the Problems]

The present apparatus 101 receives a video transmission request message from a client 103 in the HTTP protocol, the apparatus 101 converts the message into a message of a format of a server 102, and transmits the converted message to the server 102. Then, the apparatus 101 returns video data transferred from the server 102 to the client 103 in the HTTP protocol.

[Selected drawing] Figure 1

【TYPE OF DOCUMENT】 DRAWINGS



【FIG. 1】

FIG. 2

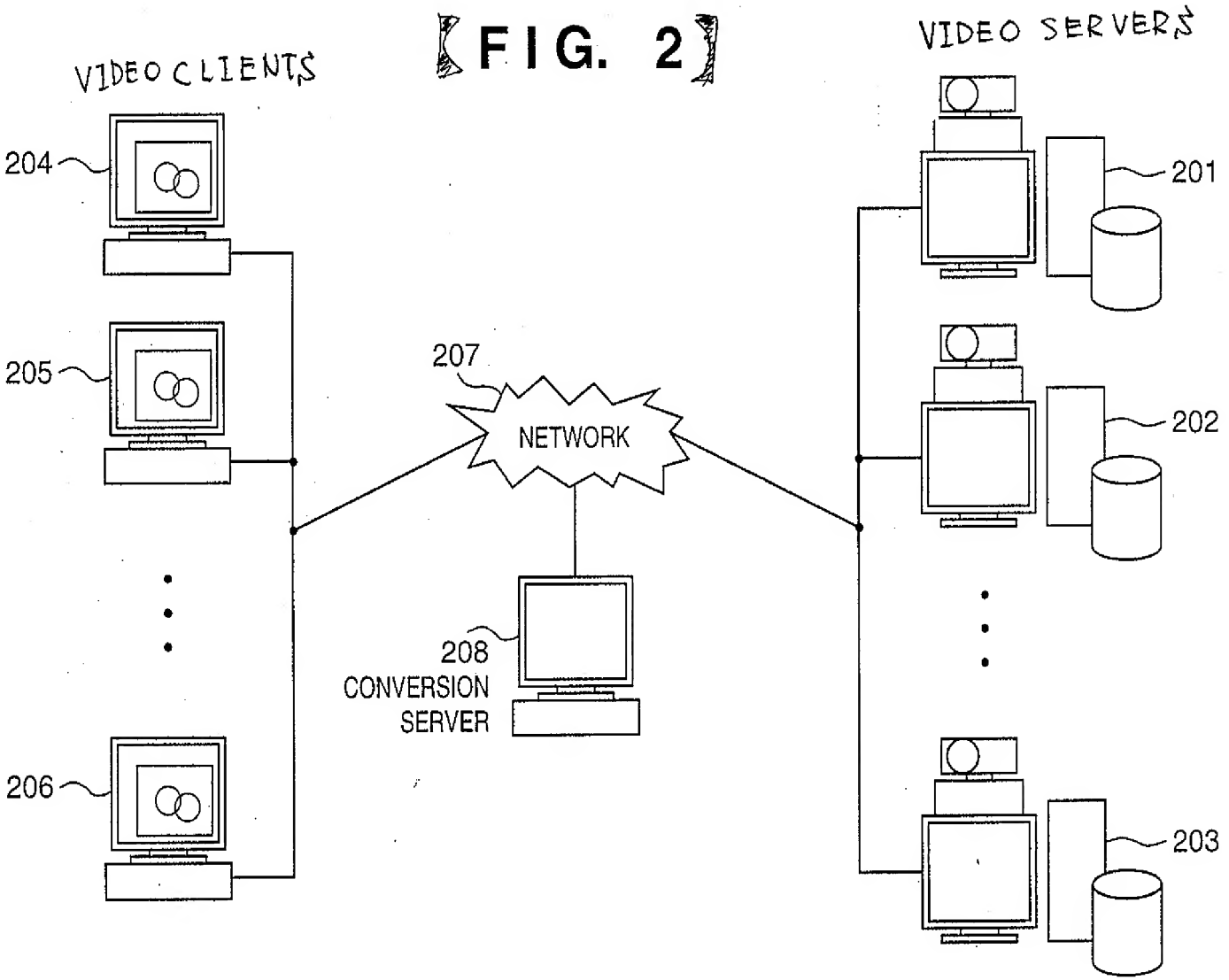
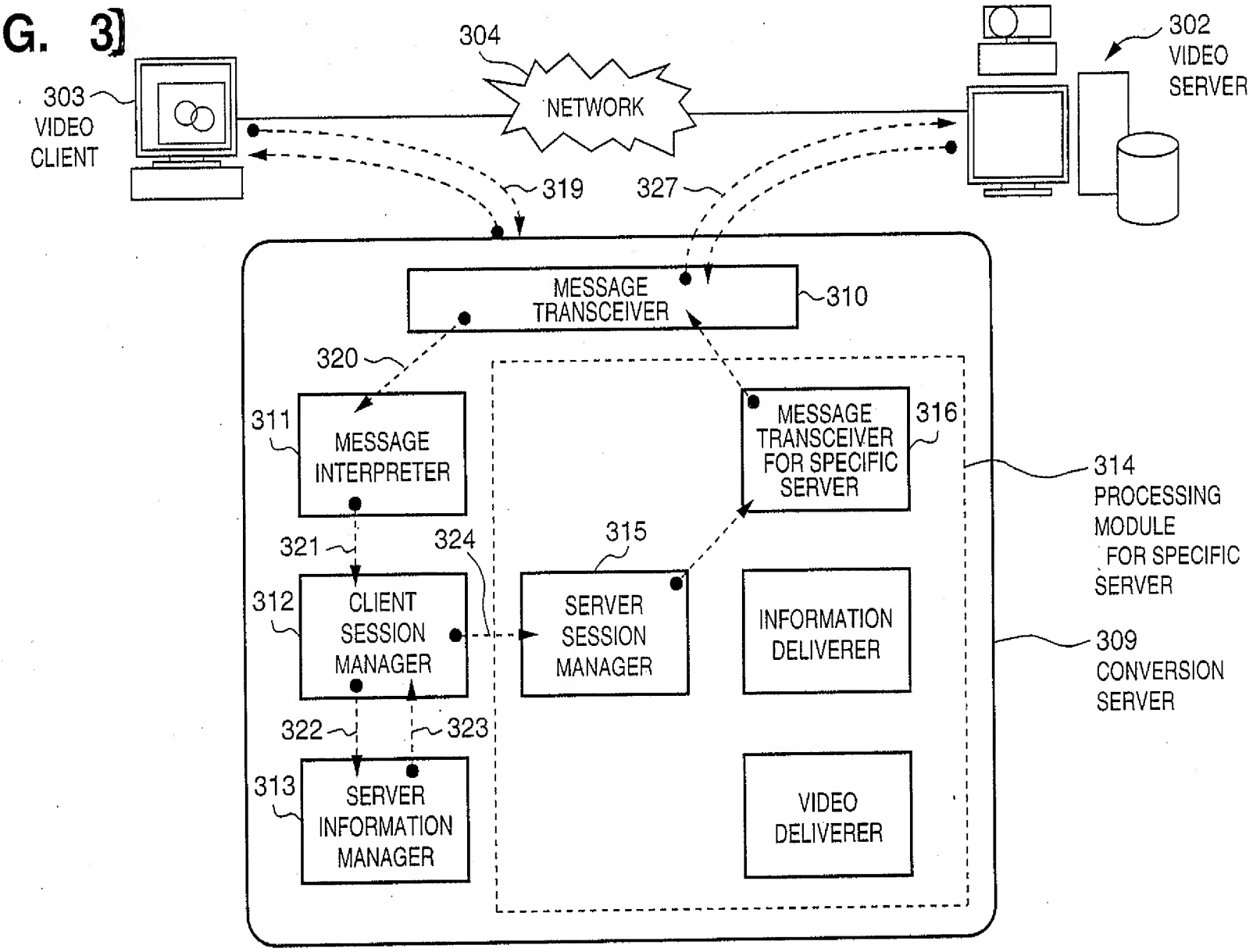

VIDEO CLIENTS
VIDEO SERVERS
204
205
206
207
NETWORK
208
CONVERSION
SERVER
201
202
203

FIG. 3


303
VIDEO
CLIENT
304
NETWORK
302
VIDEO
SERVER
319
327
MESSAGE
TRANSCEIVER
310
320
311
MESSAGE
INTERPRETER
316
MESSAGE
TRANSCEIVER
FOR SPECIFIC
SERVER
314
PROCESSING
MODULE
FOR SPECIFIC
SERVER
321
324
315
312
CLIENT
SESSION
MANAGER
SERVER
SESSION
MANAGER
INFORMATION
DELIVERER
309
CONVERSION
SERVER
322
323
313
SERVER
INFORMATION
MANAGER
VIDEO
DELIVERER

【FIG. 4】

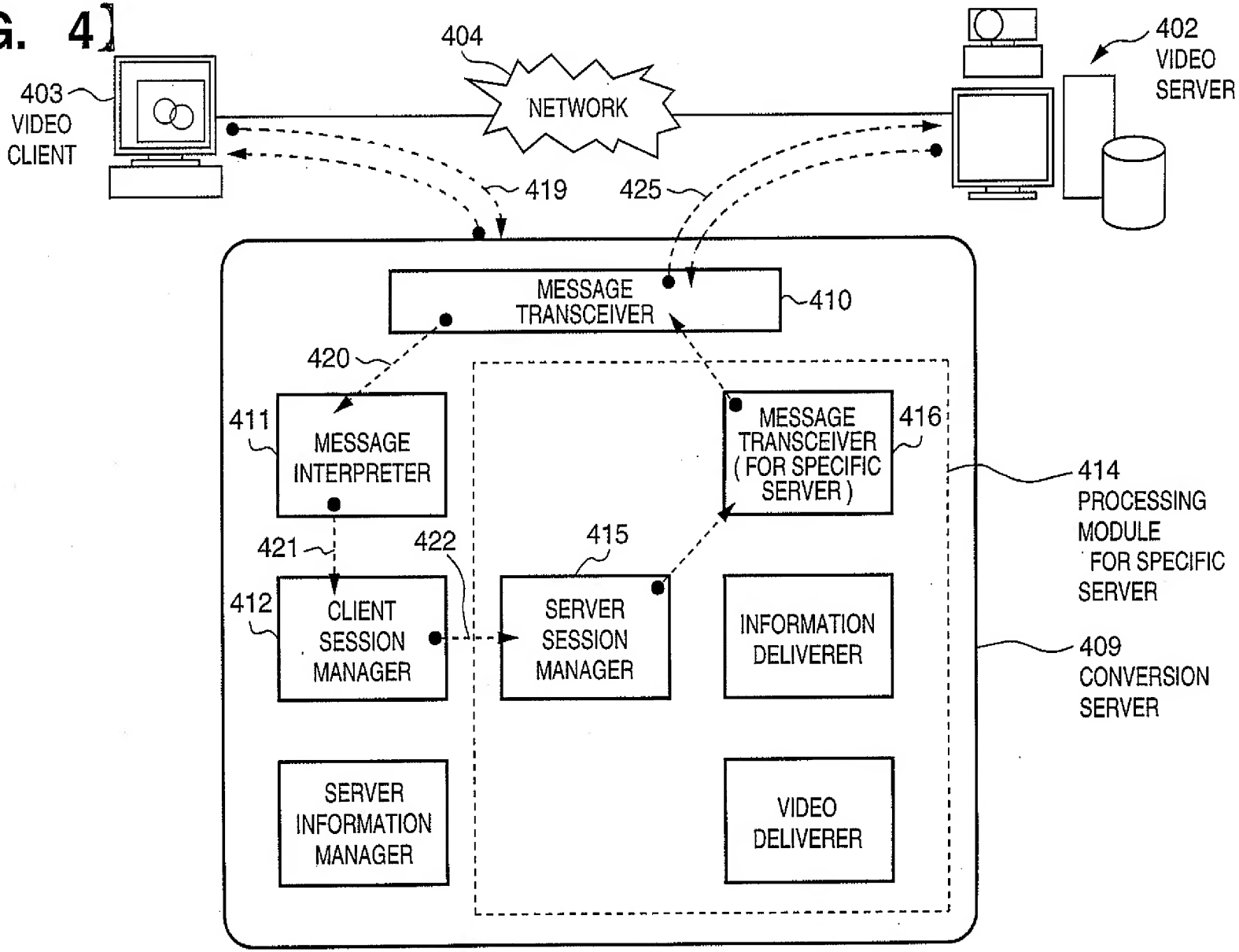
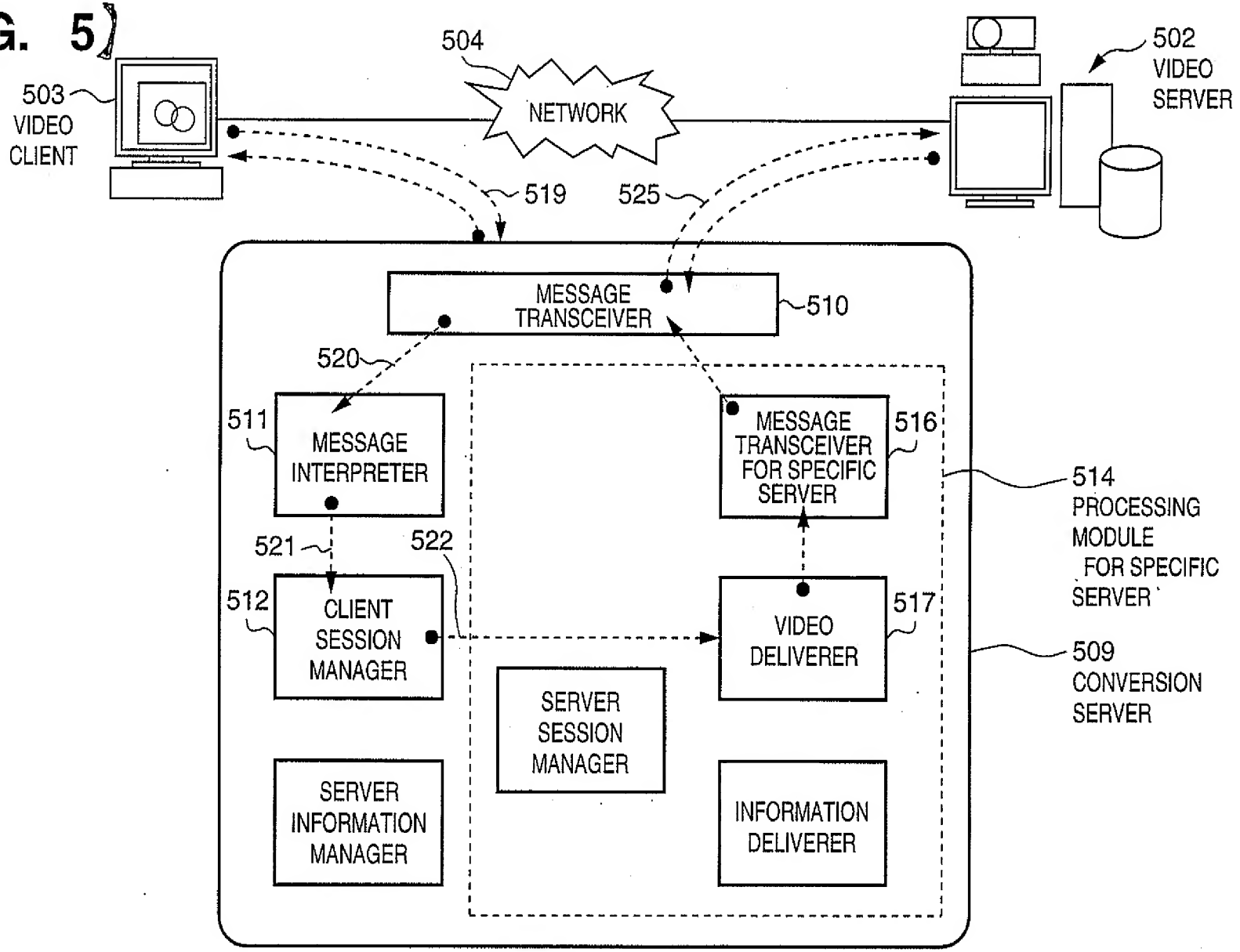
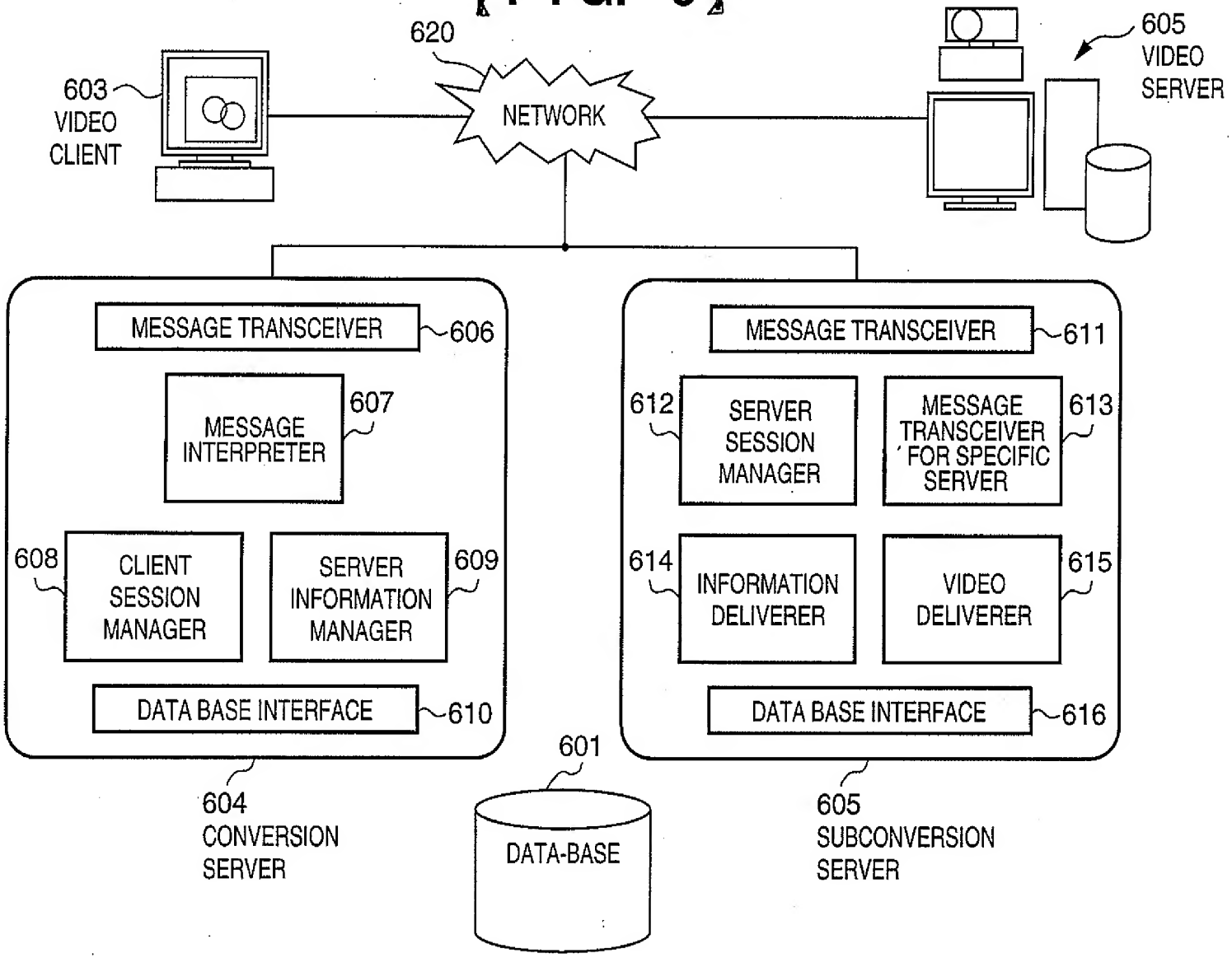

403
VIDEO
CLIENT
404
NETWORK
402
VIDEO
SERVER
419
425
MESSAGE
TRANSCEIVER
410
420
411
MESSAGE
INTERPRETER
MESSAGE
TRANSCEIVER
(FOR SPECIFIC
SERVER)
416
414
PROCESSING
MODULE
FOR SPECIFIC
SERVER
421
422
415
412
CLIENT
SESSION
MANAGER
SERVER
SESSION
MANAGER
INFORMATION
DELIVERER
409
CONVERSION
SERVER
SERVER
INFORMATION
MANAGER
VIDEO
DELIVERER

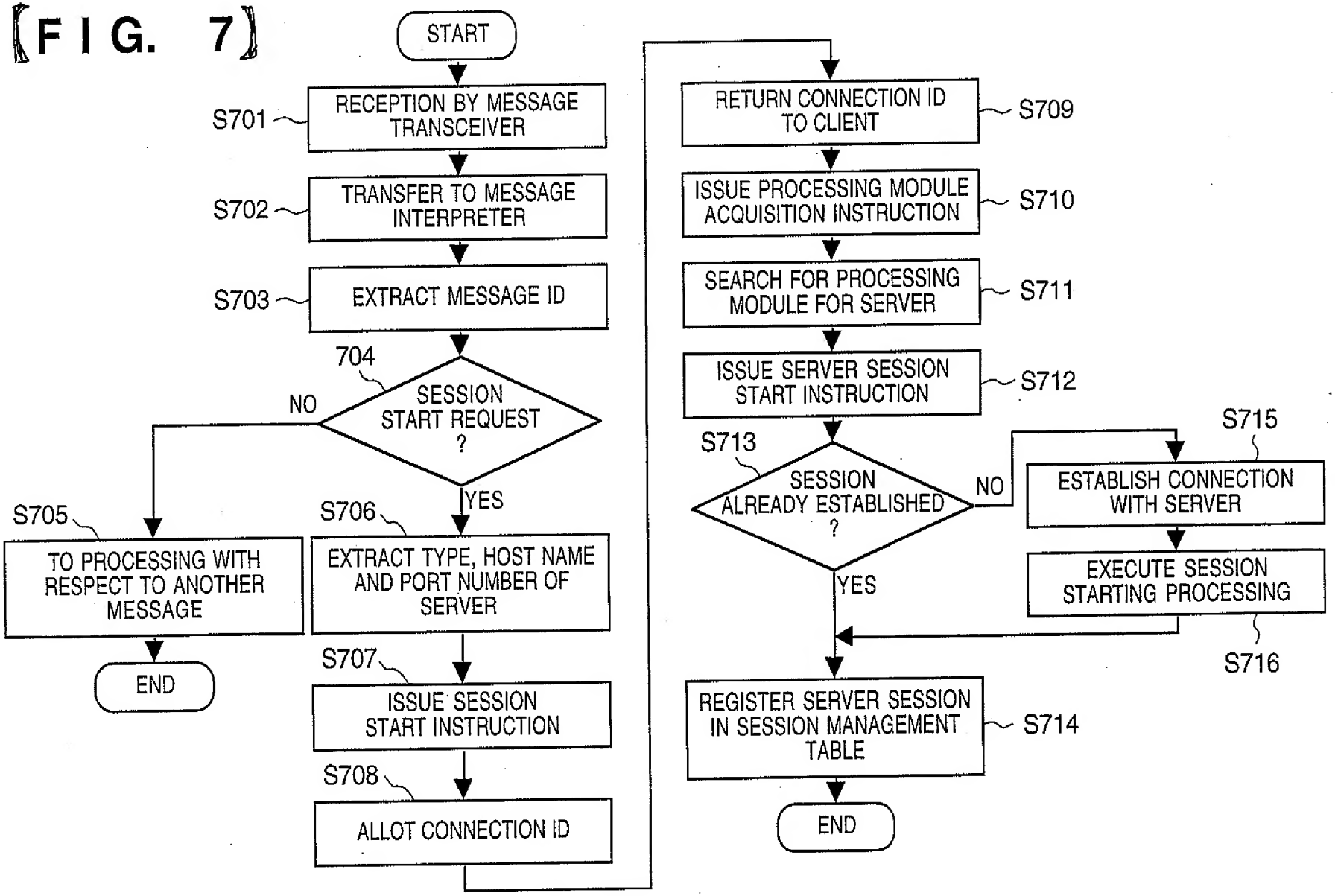
FIG. 5


503
VIDEO
CLIENT
504
NETWORK
502
VIDEO
SERVER
519
525
MESSAGE
TRANSCEIVER
510
520
511
MESSAGE
INTERPRETER
516
MESSAGE
TRANSCEIVER
FOR SPECIFIC
SERVER
514
PROCESSING
MODULE
FOR SPECIFIC
SERVER
521
522
512
CLIENT
SESSION
MANAGER
517
VIDEO
DELIVERER
509
CONVERSION
SERVER
SERVER
SESSION
MANAGER
SERVER
INFORMATION
MANAGER
INFORMATION
DELIVERER

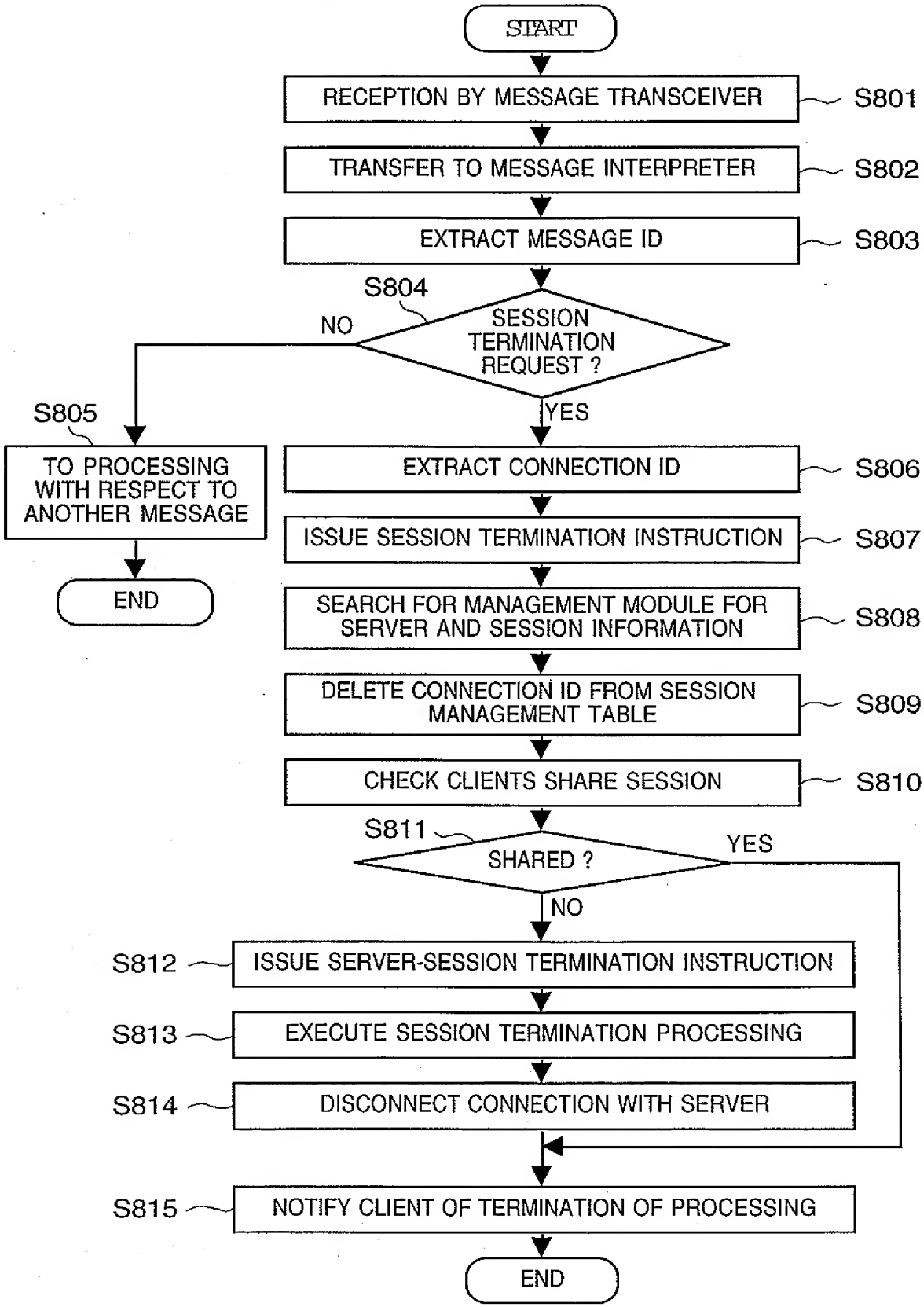
[FIG. 6]


620
NETWORK
603
VIDEO
CLIENT
605
VIDEO
SERVER
MESSAGE TRANSCEIVER
606
MESSAGE
INTERPRETER
607
608
CLIENT
SESSION
MANAGER
SERVER
INFORMATION
MANAGER
609
DATA BASE INTERFACE
610
604
CONVERSION
SERVER
601
DATA-BASE
MESSAGE TRANSCEIVER
611
612
SERVER
SESSION
MANAGER
MESSAGE
TRANSCEIVER
FOR SPECIFIC
SERVER
613
614
INFORMATION
DELIVERER
VIDEO
DELIVERER
615
DATA BASE INTERFACE
616
605
SUBCONVERSION
SERVER

【FIG. 7】


START
S701
RECEPTION BY MESSAGE TRANSCEIVER
S702
TRANSFER TO MESSAGE INTERPRETER
S703
EXTRACT MESSAGE ID
704
SESSION START REQUEST ?
NO
YES
S705
TO PROCESSING WITH RESPECT TO ANOTHER MESSAGE
END
S706
EXTRACT TYPE, HOST NAME AND PORT NUMBER OF SERVER
S707
ISSUE SESSION START INSTRUCTION
S708
ALLOT CONNECTION ID
RETURN CONNECTION ID TO CLIENT
S709
ISSUE PROCESSING MODULE ACQUISITION INSTRUCTION
S710
SEARCH FOR PROCESSING MODULE FOR SERVER
S711
ISSUE SERVER SESSION START INSTRUCTION
S712
S713
SESSION ALREADY ESTABLISHED ?
NO
YES
S715
ESTABLISH CONNECTION WITH SERVER
EXECUTE SESSION STARTING PROCESSING
S716
REGISTER SERVER SESSION IN SESSION MANAGEMENT TABLE
S714
END

【FIG. 8】


START
RECEPTION BY MESSAGE TRANSCEIVER
S801
TRANSFER TO MESSAGE INTERPRETER
S802
EXTRACT MESSAGE ID
S803
S804
SESSION TERMINATION REQUEST ?
NO
YES
S805
TO PROCESSING WITH RESPECT TO ANOTHER MESSAGE
END
EXTRACT CONNECTION ID
S806
ISSUE SESSION TERMINATION INSTRUCTION
S807
SEARCH FOR MANAGEMENT MODULE FOR SERVER AND SESSION INFORMATION
S808
DELETE CONNECTION ID FROM SESSION MANAGEMENT TABLE
S809
CHECK CLIENTS SHARE SESSION
S810
S811
SHARED ?
YES
NO
S812
ISSUE SERVER-SESSION TERMINATION INSTRUCTION
S813
EXECUTE SESSION TERMINATION PROCESSING
S814
DISCONNECT CONNECTION WITH SERVER
S815
NOTIFY CLIENT OF TERMINATION OF PROCESSING
END

[FIG. 9]

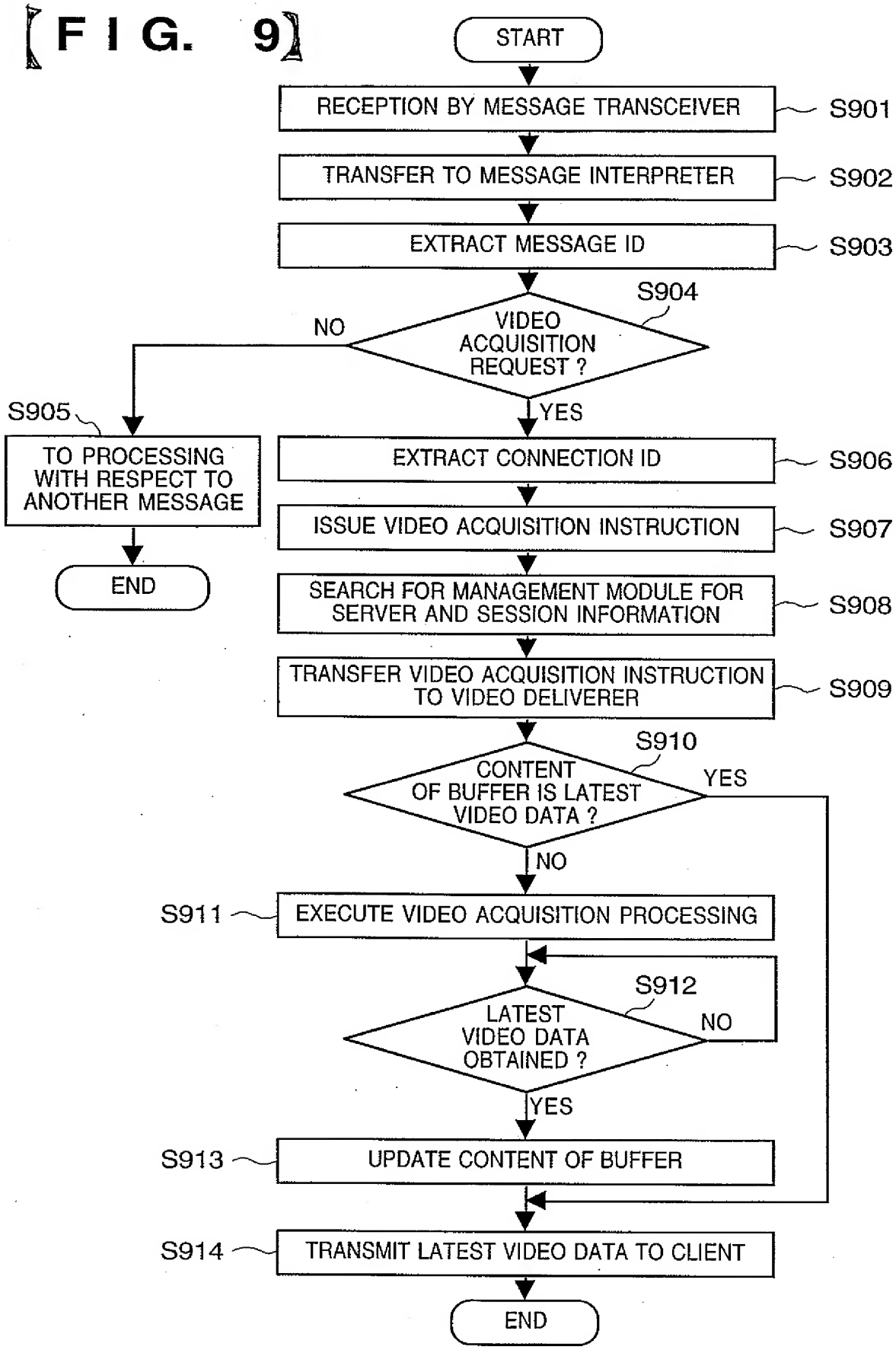
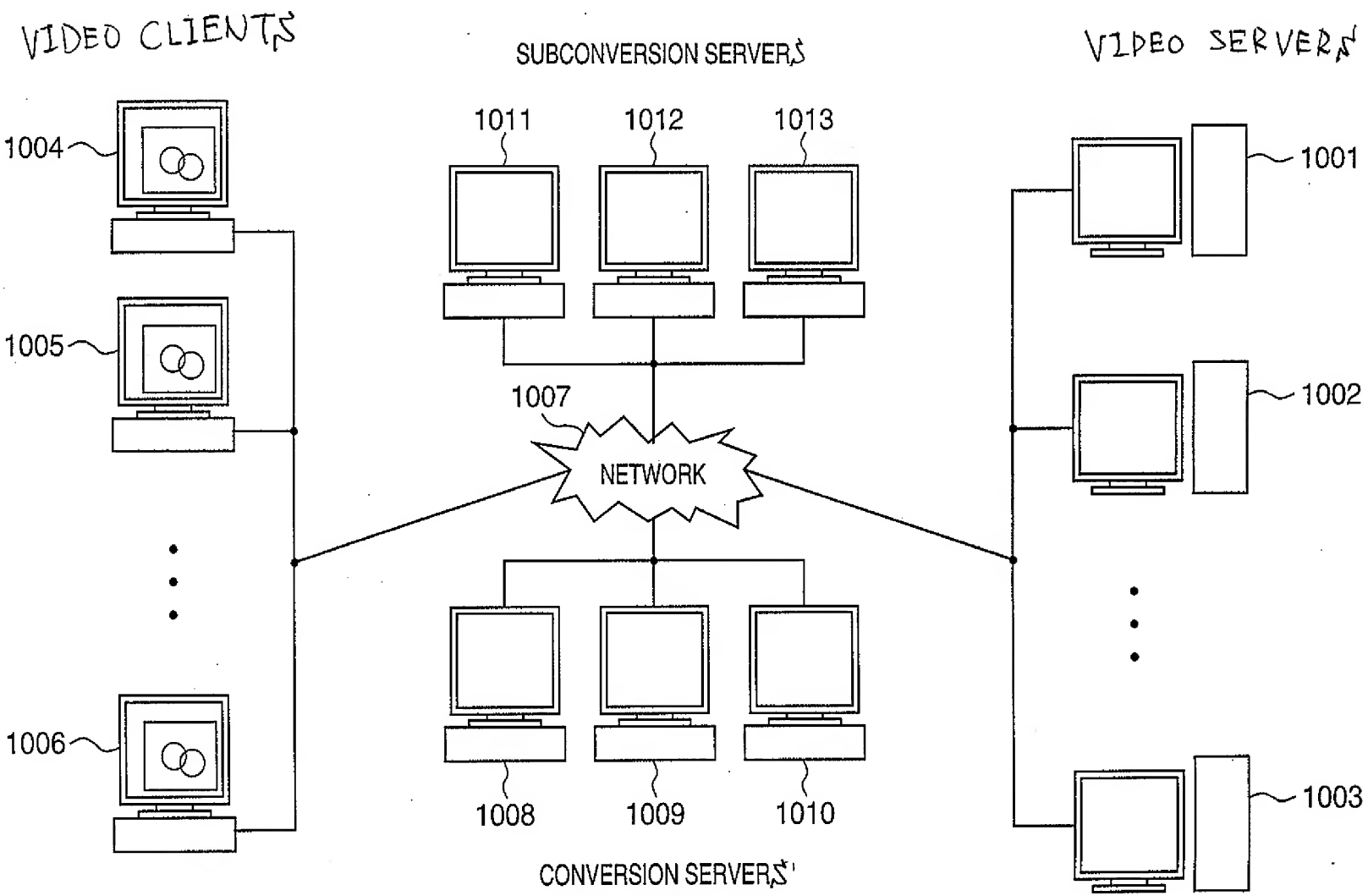

START
RECEPTION BY MESSAGE TRANSCEIVER
S901
TRANSFER TO MESSAGE INTERPRETER
S902
EXTRACT MESSAGE ID
S903
S904
VIDEO ACQUISITION REQUEST ?
NO
YES
S905
TO PROCESSING WITH RESPECT TO ANOTHER MESSAGE
END
EXTRACT CONNECTION ID
S906
ISSUE VIDEO ACQUISITION INSTRUCTION
S907
SEARCH FOR MANAGEMENT MODULE FOR SERVER AND SESSION INFORMATION
S908
TRANSFER VIDEO ACQUISITION INSTRUCTION TO VIDEO DELIVERER
S909
S910
CONTENT OF BUFFER IS LATEST VIDEO DATA ?
YES
NO
S911
EXECUTE VIDEO ACQUISITION PROCESSING
S912
LATEST VIDEO DATA OBTAINED ?
NO
YES
S913
UPDATE CONTENT OF BUFFER
S914
TRANSMIT LATEST VIDEO DATA TO CLIENT
END

FIG. 10


VIDEO CLIENTS
SUBCONVERSION SERVERS
VIDEO SERVERS
1004
1005
1006
1011
1012
1013
1007
NETWORK
1001
1002
1003
1008
1009
1010
CONVERSION SERVERS